

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

2013

Łukasz Feber

Zadání bakalářské práce

Student: **Łukasz Feber**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: New Travel.cz s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
 - c) Zvolený postup řešení zadaných úkolů
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.

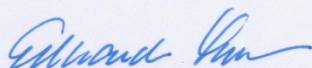
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Kot, Ph.D.**

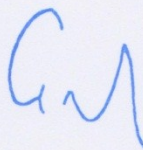
Konzultant bakalářské práce: Tomáš Černý

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

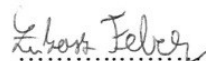


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 6.5.2013


.....
podpis studenta

Poděkování

Rád bych poděkoval Tomáši Černému a celému IT oddělení za odbornou pomoc a konzultaci při vytváření této diplomové práce.

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.“

Dne: 6.5.2013

.....
podpis zástupce

Abstrakt

V této bakalářské práci jsem popsal průběh mé odborné praxe, kterou jsem absolvoval v cestovní agentuře New Travel.cz s.r.o. Popisuji zde řešení obdržených úkolů a také způsob uplatnění teoretických znalostí získaných během studia. Hlavní náplní praxe bylo programování modulů administračního systému. Na závěr jsem zhodnotil průběh a popsal výhody absolvování odborné praxe oproti řešení standardní bakalářské práce.

Klíčová slova

Odborná praxe, New Travel.cz s.r.o., Liferay, Java, MySQL, statistiky, docházkový systém

Abstract

In this bachelor thesis I described process of my individual practice in the New Travel.cz Ltd company. I described how I had solved tasks and how I had used my theoretical knowledge learned at university. The main scope of practical experience was programming modules to administration system. In the end I sum up process of the individual practice and described its advantages.

Key words

Individual professional practice, New Travel.cz Ltd., Liferay, Java, MySQL, statistics, attendance system

Seznam použitých zkratk

Zkratka	Anglický význam	Český význam
SQL	Structured Query Language	strukturovaný dotazovací jazyk
HTML	HyperText Markup Language	značkovací jazyk pro hypertext
CSS	Cascading Style Sheets	kaskádové styly
AJAX	Asynchronous JavaScript and XML	asynchronní JavaScript a XML
XML	Extensible Markup Language	rozšiřitelný značkovací jazyk
WWW	World Wide Web	světová rozsáhlá síť

Obsah

1	Úvod	1
1.1	O firmě	1
1.2	Náplň práce.....	1
1.3	Pracovní prostředí.....	1
2	Kodérská práce	2
2.1	Grafika a rozvržení administračního systému	2
2.2	Tisk cestovních smluv	2
3	Programování v administračním systému	3
3.1	Popis	3
3.1.1	Menu.....	3
3.1.2	Uživatelské účty a role	3
3.2	Statistiky.....	3
3.2.1	Popis	3
3.2.2	Grafy.....	4
3.2.3	Prodejci.....	6
3.2.4	Destinace a cestovní kanceláře	7
3.3	Docházkový systém.....	7
3.3.1	Popis	7
3.3.2	Role v docházkovém systému	8
3.3.3	Měsíční přehled jednotlivce.....	8
3.3.4	Průchody.....	10
3.3.5	Soupiska	11
3.3.6	Evidence přesčasů	12
3.3.7	Evidence dovolené.....	12
4	Použité technologie	13
5	Teoretické vědomosti uplatněné na praxi	14
6	Scházející teoretické vědomosti na praxi	15
7	Závěr.....	16
8	Použitá literatura.....	17

1 Úvod

Vykonání individuální odborné praxe ve firmě bylo pro mě díky naskytnuté možnosti jasnou volbou. Praxi jsem absolvoval v cestovní agentuře New Travel.cz s.r.o. Ve firmě jsem byl zaměstnán jako Java programátor.

1.1 O firmě

Firmu New Travel.cz s.r.o. v roce 2006 společně založili Ing. Roman Břenek a Rostislav Urbánek. Firma se během několika let rozrostla do současných rozměrů, kdy zaměstnává přibližně 35 zaměstnanců a stala se jednou z největších a nejúspěšnějších on-line cestovních agentur v České republice. Důkazem toho je např. dosažený obrát za červenec roku 2012, který činil 40 452 192 Kč.

1.2 Náplň práce

Ve firmě jsem byl zařazen jako Java programátor. V úvodu jsem obdržel dvě kodérské úlohy. Mým zadáním bylo navrhnout a nasylovat grafické rozhraní administračního systému. Dále pak vytvořit přehlednou šablonu pro ekonomický tisk cestovních smluv. Po úspěšném vyřešení těchto úkolů jsem začal programovat moduly do administračního systému. Mými dvěma hlavními projekty se staly Statistiky a Docházkový systém. Programováním modulů do zmíněných projektů jsem strávil většinu času na praxi. Při nástupu nového zaměstnance do firmy jsem byl občas pověřen přípravou PC pro uživatele.

1.3 Pracovní prostředí

Po nástupu do firmy mi byl přiřazen dostatečně výkonný počítač, na kterém jsem pracoval po dobu celé praxe. Musel jsem se seznámit s programy SQL-Front a Tortoise SVN, které jsem používal při vývoji projektů.

2 Kodérská práce

2.1 Grafika a rozvržení administračního systému

Firma se na přelomu let 2011/2012 rozhodla pro přechod na nový administrační systém z důvodu omezených možností rozšíření stávajícího administračního systému. V době mého nástupu na praxi již byl nový administrační systém ve vývoji, nebyl však v ostrém provozu. Nejdůležitější část systému – zpracování objednávek – byla již po funkční stránce navržena, bylo však třeba přestylovat a rozvrhnout uživatelské rozhraní, což byl můj první úkol na praxi.

Požadavkem bylo použití zelené barvy a tak prvním krokem bylo zvolení různých odstínů této barvy pro odlišné prvky uživatelského rozhraní. K tomuto účelu skvěle posloužila webová aplikace *Color Scheme Designer*. Následně jsem si v CSS souboru vytvořil třídy, kterým jsem nastavoval typické CSS atributy jako jsou např.: šířka, výška, pozadí, odsazení, pozice. Využil jsem také některé z novějších atributů, které přišly s vydáním CSS3 – stín boxu, průhlednost a zaoblení boxu. Dalším krokem bylo upravení atributu „class“ u nejrůznějších prvků v jsp souborech.

Po aplikaci změn jsem výsledek vždy kontroloval ve třech webových prohlížečích – Google Chrome, Mozilla Firefox a Internet Explorer. Nejnovější verze vyjmenovaných prohlížečů již zobrazují webové stránky, až na výjimky, identicky, a proto jsem řešením problému tohoto typu netratil čas.

Během realizace tohoto úkolu jsem si osvěžil a také rozšířil znalosti jazyků HTML, CSS a JavaScript.

2.2 Tisk cestovních smluv

Mým druhým úkolem na praxi bylo nastavení tisku cestovních smluv. Přímý tisk detailu cestovní smlouvy z webového prohlížeče, jak jej uživatel vidí v administračním systému, by byl pro zaměstnance nepřehledný a pro firmu neekonomický. Požadavkem bylo vytvořit šablonu vhodnou pro tisk, tzn. přehledně a úsporně rozmístit důležité informace o objednávce, a vyvarovat se barev a webových prvků jako jsou např. tlačítka, select boxy a textová pole. Všem těmto požadavkům se mi povedlo vyhovět. Údaje o objednávce potřebné k tisku byly obsaženy v datech, která se načítají při rozkliknutí objednávky v administračním systému, a tak jsem pouze pomocí jazyků HTML a CSS rozvrhl a nastyloval šablonu pro tisk.

3 Programování v administračním systému

3.1 Popis

Programováním modulů do administračního systému jsem strávil většinu času na praxi. Systém je postaven na frameworku Liferay. Tento systém využívají k různým účelům téměř všichni zaměstnanci firmy. Mými dvěma hlavními projekty se staly prodejní, účetní a marketingové statistiky a docházkový systém.

3.1.1 Menu

Hlavní menu obsahuje tyto kategorie:

- Prodej – zde se zpracovávají všechny příchozí objednávky, které procházejí různými stavy až do konečného – Uzavřeno
- Účetnictví – tato kategorie slouží účetním pro vyřizování plateb
- Kontakty – zde jsou uvedeny kontakty na zaměstnance firmy

Menu jsem v průběhu praxe rozšířil o své dva projekty – Statistiky a Docházka.

3.1.2 Uživatelské účty a role

Každý zaměstnanec má svůj vlastní uživatelský účet, přes který se může přihlásit. V závislosti na přiřazených rolích má pak každý uživatel různá práva.

Mezi nejběžnější role v systému patří:

- Prodejce – vidí a je schopen editovat pouze jemu přiřazené objednávky a nemá přístup k žádným administračním modulům
- SuperVisor – vidí a je schopen editovat všechny objednávky, nemá však přístup k administračním modulům
- Účetní – má přístup do kategorie Účetnictví
- Administrátor – je oprávněn k editaci veškerého obsahu, má přístup do kontrolního panelu

3.2 Statistiky

3.2.1 Popis

Firma denně vyřizuje velké množství objednávek, čímž je vytvářen velký objem dat. Statistiky jsou tedy z důvodu přehledu o prodeji zájezdů nezbytně nutné. Jsou také velkým pomocníkem při tvorbě marketingové strategie a zvyšování efektivity prodeje.

Všechny příchozí objednávky jsou ukládány do databáze, jejíž tabulka Objednávky dnes čítá několik desítek tisíc záznamů. Tato tabulka obsahuje veškeré potřebné informace jako např.: datum a

čas vložení objednávky, identifikační číslo (dále jen id) země, id lokality, id prodejce, id cestovní kanceláře, typ stravování, typ dopravy, odkud přišla objednávka a zda se jedná o first minute. Tyto informace jsem používal pro tvorbu statistik.

Kategorie statistiky obsahuje následující moduly:

- Grafy – měsíční přehled prodeje zájezdů znázorněný pomocí dvou grafů
- Prodejci – statistiky prodeje zájezdů seříděné dle prodejců
- Destinace – statistiky prodeje zájezdů seříděné dle zemí a jejich lokalit
- Cestovní kanceláře – statistiky prodeje zájezdů seříděné dle cestovních kanceláří
- Důvod storna – statistiky zobrazené v podobě koláčového grafu, které zobrazují četnost různých důvodů stornování zájezdů (vyprodáný hotel, klient nemá o rezervaci zájem, klient koupil zájezd jinde..)
- Vytížení prodejců – statistiky seříděné dle prodejců a jejich vytíženosti

3.2.2 Grafy

V horní části tohoto modulu se podobně jako u všech dalších statistik nachází zaškrťovací políčka a list boxy sloužící pro určování parametrů vyhledávání a také tlačítko *Vyhledat* pro aktualizaci zobrazení. Pro vyhledávání jsou použity následující parametry:

Seznamy (list boxy):

- Rok
- Měsíc
- Cestovní kancelář
- Prodejce
- Země
- Lokalita
- Typ zájezdu

Zaškrťovací políčka (check boxy):

- Odkud objednávka přišla (možnost zaškrtnout více způsobů)
 - o User (vysvětleno níže)
 - o Web
 - o Email
 - o Telefonicky
 - o Pobočka
 - o Poradce

Pod Vyhledávačem se pak nacházejí samotné grafy. Ke grafickému vykreslení jsem použil HTML5/JavaScriptovou knihovnu *HighCharts*. Vycházel jsem z modelu „basic column“, který jsem

upravil podle vlastních potřeb v online web editoru *jsFiddle*. Každá hodnota na ose x odpovídá jednomu dni v měsíci. U každého dne se nacházejí dva sloupce. První sloupec představuje počet příchozích objednávek v daném dni a zvoleném roce. V druhém sloupci je zobrazeno totéž, avšak za minulý rok.

V horním grafu se zobrazuje počet příchozích objednávek bez ohledu na to, zda byly úspěšně prodány. Dolní graf vykresluje počet dokončených objednávek.

Ve výchozím zobrazení je u parametru „Odkud objednávka přišla“ zaškrtnuta speciální varianta User. Tato varianta byla vytvořena za účelem zpřesnění výpočtu efektivity prodeje. Její zaškrtnutí způsobuje, že v horním grafu se zobrazují pouze objednávky, které přišly z webu, telefonicky anebo byly vyřizovány osobně na pobočce. Způsoby obdržení Email a Poradce nejsou v horním grafu obsaženy, jelikož se ve většině případů jedná pouze o dotazy týkající se zájezdu. V dolním grafu se u této varianty zobrazují všechny typy obdržení objednávky.

Zde je ukázka SQL dotazu, který je použit pro výběr dat z databáze. Dotaz vybere počet zájezdů do Egypťské (countries_id = 8) lokality Dahab (localities_id = 3), které v únoru (month(o.sold) = 2) roku 2013 (year(o.sold) = 2013) úspěšně prodal prodejce, jehož identifikační číslo je 13246 (o.user_id = 13246). Data se následně seřadí do jednotlivých dnů měsíce (GROUP BY day_num) a také seřadí (ORDER BY day_num).

```
SELECT
    day(o.sold) AS day_num,
    count(o.id) AS count_orders
FROM orders o
WHERE
    month(o.sold) = '2' AND
    year(o.sold) = '2012' AND
    o.user_id = '13246' AND
    o.suppliers_id = '2' AND
    o.countries_id = '8' AND
    o.localities_id = '3'
GROUP BY day_num
ORDER BY day_num;
```

Zdrojový kód č. 1: SQL dotaz pro výběr dat do modulu Grafy

Výsledná data jsou následně uložena do listu „statisticsList“, který je z kontroléru předán do view pomocí následujícího příkazu:

```
setAttribute(„data“, statisticsList);
```

Zdrojový kód č. 2: Předání dat z kontroléru do view

View představuje soubor „view.jsp“, do kterého je zasazen JavaScriptový kód pro inicializaci *HighCharts* grafů:

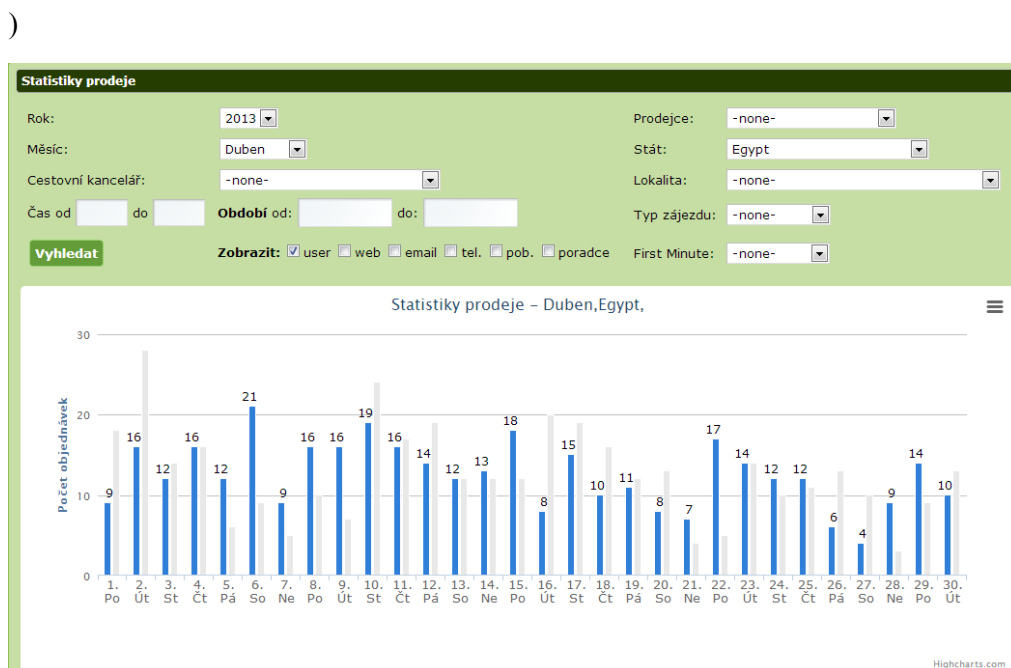
```
var chart;
chart = new Highcharts.Chart({
    chart: {
        renderTo: 'container',
        type: 'column'
```

```

    },
    xAxis: {
        categories: [
            <c:forEach items="${data}" var="d" varStatus="status">
                '${d.day_num}. <br />${d.day_name}' <c:if test="${not
status.last}">, </c:if>
            </c:forEach>,
        ],
        series: [{
            name: 'Objednávky',
            data: [<c:forEach items="${data}" var="d"
varStatus="status">
                ${d.count_orders}<c:if test="${not
status.last}">, </c:if>
            </c:forEach>],
        }]
    }
});

```

Zdrojový kód č. 3: Kód pro inicializaci HighCharts grafů



Obrázek č. 1: Modul „Statistiky“ v administracním systému

3.2.3 Prodejci

Tento modul byl naprogramován za účelem získání přehledu o jednotlivých prodejcích a jejich úspěšnosti prodeje. Tabulka obsahuje seznam prodejců a počet jim přiřazených objednávek, počet aktuálně rozpracovaných objednávek, počet úspěšně/ neúspěšně vyřízených objednávek a v neposlední řadě efektivitu. Efektivita je rovná podílu počtu úspěšně vyřízených objednávek a počtu přiřazených objednávek. Počet prodaných zájezdů a efektivita má samozřejmě velký vliv na výši odměny prodejce.

Data v tabulce můžeme, stejně jako v modulu *Grafy*, omezovat různými filtry jako je např. časové období nebo typ objednávky.

3.2.4 Destinace a cestovní kanceláře

Tyto dva moduly se svým rozvržením a logikou velmi podobají. V případě modulu Destinace jde o statistiky prodeje zájezdů do jednotlivých zemí. Modul *Cestovní kanceláře* mezi sebou porovnává jednotlivé cestovní kanceláře. U obou modulů nalezneme sloupce jako např.: počet příchozích objednávek, počet úspěšně vyřízených objednávek, efektivita a obrát.

U modulu Destinace bylo navíc požadavkem, aby se po kliknutí na název země zobrazily statistiky cestovních kanceláří dané země. Využil jsem technologii AJAX, aby při zobrazení detailu země nedocházelo k obnovení stránky.

Zobrazená data si můžeme pomocí *Vyhledávače* různými způsoby vyfiltrovat a seřadit.

3.3 Docházkový systém

3.3.1 Popis

V dnešní době již téměř každá větší firma vlastní svůj docházkový systém. Každý zaměstnanec vlastní svůj čip, který slouží k monitorování příchodů a odchodů z práce. Díky programům se zaměstnanec může kdykoliv podívat, jaký je jeho aktuální docházkový stav. Programy se také starají o výpočet odpracovaných hodin za aktuální měsíc, přesčasů či zbývajících dnů dovolené. Firma New Travel.cz s.r.o. se s rostoucím počtem zaměstnanců také rozhodla pro zakoupení tohoto technického řešení. Konkrétně se jednalo o docházkový terminál *REX* od firmy *Z-WARE*, která se zaměřuje jak na hardware, tak i na software. Z důvodu jiné softwarové platformy (Windows) se firma rozhodla zakoupit pouze samotné docházkové zařízení s čipy a naprogramovat si vlastní software, což se stalo mým dalším významným projektem.

Docházkový systém se skládá z následujících modulů:

- Docházka – denní přehled zvoleného zaměstnance za vybraný měsíc
- Průchody – výpis všech záznamů zvoleného zaměstnance za vybraný měsíc
- Soupiska(HPP) – měsíční výpis všech zaměstnanců s pracovním poměrem HPP
- Soupiska(DPP) – měsíční výpis všech zaměstnanců s pracovním poměrem DPP
- Evidence přesčasů – seznam zaměstnanců a informace o dosažených přesčasech za jednotlivé měsíce
- Evidence dovolené – seznam zaměstnanců a informace o možnosti čerpání dovolené

Pro účely docházkového systému jsem v databázi administračního systému vytvořil následující tabulky s prefixem Att (tj. Attendance):

- Att_passages – tabulka obsahující všechny záznamy/ průchody, které jsou vkládány díky triggeru aplikovanému na tabulku v databázi docházkového systému
- Att_roster – tabulka obsahující informace o docházce zaměstnanců za jednotlivý měsíc – počet odpracovaných hodin, počet vyčerpané dovolené, přesčasy atd.
- Att_shifts – tabulka obsahující záznamy o dovolených, školeních, nemocenské atd.
- Att_holiday – tabulka obsahující informace jako jsou např.: počet dní dovolené převedených z minulého roku a nárok na dovolenou v aktuálním roce

3.3.2 Role v docházkovém systému

Pro docházkový systém jsem v administračním rozhraní vytvořil tyto role:

- Docházka – uživatel má možnost prohlížet vlastní docházku a také svou evidenci přesčasů a dovolené
- Docházka Editor – uživatel smí zobrazit i editovat docházku prodejcům a brigádníkům, má přístup k soupiskám
- Docházka Master – uživatel smí zobrazit i editovat docházku kohokoliv, má přístup k soupiskám

3.3.3 Měsíční přehled jednotlivce

Měsíční přehled jednotlivce (tj. modul *Docházka*) slouží především zaměstnancům, aby měli možnost si zkontrolovat, zda byly všechny příchody, odchody a polední pauzy správně zaznamenány.

Uživatel, který je v roli *Docházka Editor* nebo *Docházka Master*, má navíc u každého dne tlačítko Editovat, pomocí kterého může přidat, smazat nebo změnit jakýkoliv záznam. Každá z možností – dovolená, nemoc, práce z domu atd. má přiřazený svůj vlastní select box, ve kterém je možno navolit počet hodin v daném dni. SQL dotaz pro vložení 4 hodinové dovolené ke dni 21. března 2013, kterou absolvoval uživatel s identifikačním číslem 12345, by vypadal následovně (v číselníku type_shift má dovolená id č.1):

```
INSERT INTO Att_shifts (date, userId, type_shift_id, hours) VALUES  
(,2013-03-21`, 12345, 1, 4);
```

Zdrojový kód č. 4: SQL dotaz pro vložení záznamu o dovolené

Těmto privilegovaným uživatelům se taky navíc v horní části modulu zobrazuje select box, pomocí kterého si můžou navolit prodejce, jehož docházku chtějí zobrazit.

Aby systém dokázal rozlišit, v jaké roli je aktuálně přihlášený uživatel, postupujeme následovně: v kontroléru si do atributu „isAttendanceEditor“ uložíme funkci „isAttendanceEditor“, která nám v závislosti na aktuálně přihlášeném uživateli a jeho přiřazeným rolím odpoví, zda je v roli *Attendance Editor*:

```
setAttribute("isAttEditor", core.Portal.isAttEditor(getRequest()));
```

Zdrojový kód č. 5: Předání informace o aktuálně přihlášeném uživateli do view

Následně ve view pomocí podmínky *if* ověříme, zda se jedná o uživatele v roli *Attendance Editor* a v případě že ano, zobrazíme select box pro výběr osoby:

```
<c:if test = "${isAttendanceEditor}">  
  <div class="searchBoxSmall">  
    Osoba <p:select name="user" var="${users}" value="${search.user}"/>  
  </div>  
</c:if>
```

Zdrojový kód č. 6: Podmínka v jsp souboru

Aby se každému prodejci zobrazila právě jeho docházka, do proměnné `userId` jsem uložil identifikační číslo aktuálně přihlášeného uživatele. Proměnnou jsem pak použil jako parametr v SQL dotazu.

```
long userId = core.Portal.getUserId(getRequest());
```

Zdrojový kód č. 7: Uložení loginu aktuálně přihlášeného uživatele do proměnné

Prodejcům začíná pracovní doba vždy v 7:30, a tak bylo třeba upravit jejich záznamy v případě dřívějšího příchodu do firmy. Tento problém jsem vyřešil přidáním podmínky do triggeru, který patří k tabulce v databázi docházkového systému a jeho úkolem je kopírovat záznamy do databáze administračního rozhraní. Čas vložení záznamu se do databáze ukládá v nepřehledném formátu „20130301073015“, což odpovídá datu 1. března 2013 a času 7:30:15. Z tohoto důvodu jsem ho pomocí funkcí `STR_TO_DATE`, `CONCAT` a `SUBSTR` upravil do formátu datového typu `datetime` – „2013-03-01 07:30:15“, se kterým se mnohem lépe operuje. Zde jsou vybrány části kódu triggeru týkající se výše zmíněného problému:

```
declare v_arrivalDate datetime;
declare v_0730Date datetim;

SET v_arrivalDate = (STR_TO_DATE(NEW.PRUCHOD, '%Y%m%d%H%i%s'));
SET v_0730Date = (STR_TO_DATE(CONCAT(SUBSTR(NEW.PRUCHOD, 1, 8),
'073000'), '%Y%m%d%H%i%s'));

IF(v_arrivalDate < v_0730Date)
THEN

    INSERT INTO newtravel.Att_passages
    (chip, personalNumber, dateTime, type) VALUES
    (NEW.CIP, NEW.OSC, v_0730Date , NEW.KOMENTAR);

END IF;
```

Zdrojový kód č. 8: Kód triggeru pro docházkový systém

Docházka

Vyhledávání

Měsíc

Rok

Osoba

Březen

2013

Feber Luka

Vyhledat

3, 2013 - Feber Lukasz | Fond pracovní doby: 168 | Odpr. hod. + náhrady: 83:43

Den	Příchod	Odchod	Přerušení	Celkem	Edit.	D	NV	P	N	PzD	Š	NaV
Suma				83:43		0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.			00:00	00:00	E	0	0	0	0	0	0	0.0
2.			00:00	00:00	E	0	0	0	0	0	0	0.0
3.			00:00	00:00	E	0	0	0	0	0	0	0.0
4.	09:09	17:13	00:30	07:34	E	0	0	0	0	0	0	0.0
5.	09:07	16:11	00:49	06:15	E	0	0	0	0	0	0	0.0
6.			00:00	00:00	E	0	0	0	0	0	0	0.0
7.			00:00	00:00	E	0	0	0	0	0	0	0.0
8.	16:33	17:12	00:00	00:38	E	0	0	0	0	0	0	0.0

Obrázek 2: Modul „Docházka“ v administračním rozhraní

3.3.4 Průchody

Tento modul slouží k řešení nesrovnalostí v případě zvolení špatného typu průchodu zaměstnancem (např. místo typu „odchod na oběd“ zaměstnanec omylem zvolil „odchod“), což mohlo způsobit nekorektní zobrazení v modulu *Docházka*. Jde o jednoduchou tabulku, která obsahuje všechny průchody (čas vykonání a zvolený typ) vybraného zaměstnance za zvolený měsíc.

Data pro tabulku jsou vybrána z databáze SQL dotazem. Následující kód vybere z databáze všechny průchody uživatele s identifikačním číslem „12345“ za březen roku 2013.

```
SELECT
    personalNumber,
    DATE_TO_STR(dateTime, '%d.%M.%Y %H:%i%s'),
    type
FROM
    att_passages
WHERE
    userId = 12345 AND
    month(dateTime) = 3 AND
    year(dateTime) = 2013;
```

Zdrojový kód č. 9: SQL dotaz pro výběr dat do modulu Průchody

3.3.5 Soupiska

Tento modul slouží jako přehled pro účetní a nadřízené pro kontrolu docházky všech zaměstnanců. Tabulka je strukturou přizpůsobená pro výpočet mezd. Ke zvolenému měsíci zobrazuje počet odpracovaných hodin, počet vyčerpaných dní dovolené/ nemocenské/ práce z domu a u prodejců je také zobrazena hodnota odměny, která je vypočtena z obrátu a počtu prodaných zájezdů daného prodejce.

Soupisky jsem rozdělil do dvou modulů – jeden obsahuje seznam zaměstnanců s HPP (hlavní pracovní poměr) a druhý seznam zaměstnanců s DPP (dohoda o provedení práce). Rozdělení je z důvodu přehlednosti a také odlišného způsobu výpočtu mezd.

Veškerá data pro tyto moduly jsou uchovávána v databázové tabulce *Att_roster*. Mým původním úmyslem bylo získat data pomocí SQL dotazu, ve kterém jsem spojil tabulky *Att_passages* (obsahuje průchody) a *Att_shifts* (obsahuje data o dovolených, nemocenských apod.) a následně vypočítával sumy odpracovaných časů všech zaměstnanců firmy. Tento způsob fungoval, avšak doba zpracování SQL dotazu byla v řádu desítek sekund, a tak bylo nutné najít jiné řešení.

V databázi jsem vytvořil proceduru *Proc_Att_roster*, která vypočítává měsíční hodnoty každého zaměstnance zobrazované v modulu *Soupiska*. Následně v aplikaci administračního rozhraní jsem vytvořil novou funkci *updateRoster*, která zavolá proceduru *Proc_Att_roster* se dvěma parametry – měsíc a rok. Pokud se funkce zavolá první den v měsíci, spustí se také procedura *Proc_Att_rewards*, která nastaví předvoleným zaměstnancům odměny:

```
public final void updateRoster() throws SQLException {
    Database write = App.getConWrite();
    int month = getMonth();
    int year = getYear();
    int dayOfMonth = getDayOfMonth();
    if (dayOfMonth == 1) {
        write.update("CALL Proc_Att_rewards(" + month + "," + year + ")");
    }
    write.update("CALL Proc_Att_roster(" + month + "," + year + ")");
}
```

Zdrojový kód č. 10: Funkce volající uloženou proceduru Proc_Att_roster

Data v tabulce *Att_roster* musí být aktuální, proto je třeba tuto funkci denně volat. K tomu, aby se tento proces nemusel provádět ručně, jsem využil Linuxovou démon aplikaci Cron. Díky tomuto softwaru, jež neustále běží na serveru, jsem pomocí vložení následujícího řádku kódu do konfiguračního souboru zajistil, aby se funkce *updateRoster* automaticky zavolala každý den ve 23:00.

```
0 23 * * * java -jar /app/NTApp.jar attendanceMonth >>
/app/log/attendanceMonth
```

Zdrojový kód č. 11: Kód z konfiguračního souboru aplikace Cron

Soupiska HPP

Vyhledávání

Měsíc Rok Zodp. osoba

Březen 2013 -none-

Vyhledat

3, 2013

Jméno	PP	Zodp. Osoba	Prac. sv. hod	fond hod	Docházka hod	Odpr. hodiny hod	z odpr.doby sob. hod	sv. hod	Přesč. hod	NÁHRADY				Osobní příplatek Kč	Odměny Kč	Srážky Kč
										D dny	N dny	NV hod	P hod			
Balon Sebastian	HPP	MŘ	0	168	169	168	0	0	0	0.0	0	0	0	0 Kč	0 Kč	0 Kč E
Břenek Roman	HPP	MŘ	0	168	168	168	0	0	0	0.0	0	0	0	0 Kč	0 Kč	0 Kč E
Břenková Zuzana	HPP	MŘ	0	168	165	160	0	0	0	1.0	0	0	0	0 Kč	0 Kč	0 Kč E
Čechová Zuzana	HPP	JM	0	168	179.5	164	0	0	0	0.0	0	0	4	5 000 Kč	4 000 Kč	0 Kč E
Čížmarová Denisa	HPP	VJ	0	168	0	0	0	0	0	0.0	21	0	0	3 000 Kč	0 Kč	0 Kč E
Čolasová Natálie	HPP	VJ	0	168	177	168	0	0	0	0.0	0	0	0	3 000 Kč	7 816 Kč	0 Kč E
Gořalková Marcela	HPP	JM	0	168	176	168	0	0	0	0.0	0	0	0	5 000 Kč	4 000 Kč	0 Kč E
Grygar David	HPP	MŘ	0	168	0	0	0	0	0	0.0	0	0	0	0 Kč	0 Kč	0 Kč E
Horká Renata	HPP	VJ	0	168	186.5	168	12	0	0	0.0	0	0	0	3 000 Kč	7 294,05 Kč	0 Kč E

Obrázek 3: Modul „Soupiska HPP“ v administračním rozhraní

3.3.6 Evidence přesčasů

Jak již název modulů napovídá, jedná se o přehled přesčasů zaměstnanců, který je zobrazený v tabulce. Data jsou čerpána z databázové tabulky *Att_roster*, ve které se nachází sloupec overtime (přesčas). Od součtu přesčasových hodin dosažených zaměstnancem za jednotlivé měsíce je odečten daný počet přesčasových hodin, kterých musí zaměstnanec v průběhu jednoho roku dosáhnout, aby měl nárok na proplacení přesčasů. Pokud je výsledná hodnota kladná, je rovná počtu hodin, které bude mít zaměstnanec proplacené. Pro vypočtení těchto hodnot jsem využil SQL funkce *TIMEDIFF* a *ADDDTIME*.

3.3.7 Evidence dovolené

V tomto modulu se nachází tabulka, která zobrazuje veškeré potřebné informace týkající se dovolené zaměstnanců – počet dní převedených z minulého roku, nárok na dovolenou, zůstatek dovolené a počet vyčerpaných dní za jednotlivé měsíce. Údaje jsou čerpány z databázových tabulek *Att_shifts* a *Att_holiday*. Z tabulky *Att_shifts* získávám počet vyčerpaných dní dovolené za jednotlivé měsíce a v tabulce *Att_holiday* jsou uloženy informace o počtu převedených dní dovolené z minulého roku a také nárok na dovolenou v aktuálním roce. Jelikož se dovolená začala evidovat až na začátku roku 2013, bylo nutné počet dní dovolené převedených z minulého roku zadat každému zaměstnanci ručně. Následující rok již však díky vytvořené proceduře vše proběhne automaticky.

4 Použité technologie

Během grafických a strukturálních úprav administračního systému jsem využil technologie HTML, CSS a JavaScript.

- HTML – HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext. Je hlavním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. [1]
- CSS – Kaskádové styly (anglicky Cascading Style Sheets) je jazyk pro popis způsobu zobrazení webových stránek napsaných v jazycích HTML, XHTML a XML. [1]
- JavaScript – JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků. [1]

Jelikož administrační systém běží na podnikovém portálu Liferay, který je založený na jazyce Java, nebylo možné přemýšlet nad volbou programovacího jazyka. Pro výběr dat z databáze, tvorbu triggerů a procedur jsem využil jazyk SQL.

- Liferay – Liferay Portal je bezplatný open-source podnikový portál, který je založený na jazyce Java a distribuovaný pod licencí GNU Lesser General Public Licence a dalšími proprietárními licencemi. Umožňuje správu dat, aplikací a procesů z jednoho centrálního uživatelského rozhraní. Portál se skládá z jednotlivých funkčních jednotek, které se nazývají portlety. [3]
- Java – Java je objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems a představila 23. května 1995. Java je jedním z nejpoužívanějších programovacích jazyků na světě. Podle Tiobe indexu je Java nejpoužívanější programovací jazyk. Dne 8. května 2007 Sun uvolnil zdrojové kódy Javy (cca 2,5 miliónů řádků kódu) a Java bude dále vyvíjena jako open source. [1]
- MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. [2]
- SQL – je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. SQL je zkratka anglických slov Structured Query Language (strukturovaný dotazovací jazyk). [1]

5 Teoretické vědomosti uplatněné na praxi

Během praxe jsem uplatnil především základní znalosti jazyků Java a SQL, které jsem získal absolvováním těchto předmětů:

- Programovací jazyky (PJ1)
- Úvod do databázových systémů (UDBS)
- Databázové a informační systémy (DAIS)

Uplatnil jsem také znalosti jazyků HTML, CSS a JavaScript, které jsem díky tvorbě vlastních webových stránek ovládal již při nástupu na praxi.

6 Scházející teoretické vědomosti na praxi

Mým největším problémem bylo nedostatečné soužití a zkušenosti s programovacími jazyky Java a SQL, což mě v úvodu praxe mírně brzdilo. S postupem času jsem si tyto technologie osvojil a téměř se zbavil problémů týkajících se nekorektní syntaxe těchto jazyků. Další ze scházejících dovedností byla správa Linux serveru. S pomocí zkušenějších kolegů jsem však již po krátké době zvládal základní operace.

7 Závěr

Absolvování odborné praxe pro mě bylo velkým přínosem. Zdokonalil jsem si své odborné znalosti a seznámil se s pracovním prostředím velké firmy. Naučil jsem se také navrhnout a zdůvodnit použití jiného způsobu řešení úkolu zadaného klientem, který se nepohybuje ve sféře informačních technologií. Díky řešení reálných problémů jsem získal zkušenosti, které bych v případě řešení klasické bakalářské práce nenabyl.

8 Použitá literatura

- [1] Internetové stránky Wikipedia.org – [online]
URL: <<http://www.wikipedia.org>> [cit. 20.4.2013]
- [2] Internetové stránky MySQL – [online]
URL: <<http://www.mysql.com>> [cit. 20.4.2013]
- [3] Internetové stránky Liferay Inc. – [online]
URL: <<http://www.liferay.com>> [cit. 20.4.2013]